Experiment #3: Digital Output Signal Conditioning

The outputs of a microcontroller are used to control the status of output field devices. Output devices are those devices that do the work in a process-control application. They deliver the energy to the process under control. A few common examples include motors, heaters, solenoids, valves, and lamps. The low-power output capability of the BASIC Stamp (or any microcontroller) prevents it from providing the power required by these loads. With proper signal conditioning, the BASIC Stamp can control power transistors, thyristors, and relays. These are the devices that can deliver the load current and voltage demands of the field devices. In some applications, you may use a BASIC Stamp output to communicate with another microcontroller or electronic circuit. There may be compatibility issues of different logic families, separate power supplies, or uncommon grounds that require special consideration. The focus of this experiment is to present some of the signal conditioning techniques used to interface your BASIC Stamp to output field devices.

Appropriate signal conditioning design begins with a brief look at the characteristics and limitations of the BASIC Stamp's outputs. The output of the BASIC Stamp is considered "standard TTL" level. As we discussed in Experiment #2, this means it can switch between logic high of approximately 5 volts or logic low of nearly 0 volts. According to the BASIC Stamp's datasheet, each output can sink 25 mA and source 20 mA of current. Relating this to the partial diagram in Figure 3.1, notice how the load can be connected. In Figure 3.1a, the load is wired from the output pin to ground. When you set an output pin high, five volts appear across the load resistor (RL). Load current will flow from ground through the resistor and into the output pin. This is the current source mode, and the BASIC Stamp can deliver a maximum of 20 mA to the load.

Figure 3.1: BASIC Stamp Output Pin Current Capability

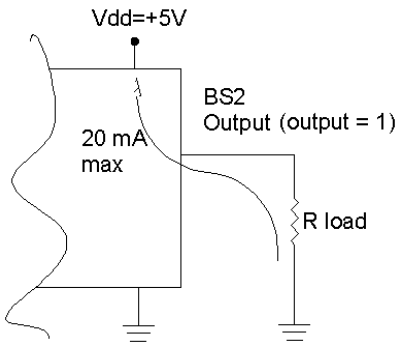


Figure 3.1a: Current Source

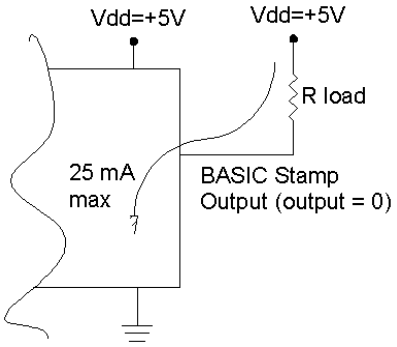


Figure 3.1b: Current Sink

In Figure 3.1b, the load is between the output pin and the +5-volt Vdd supply. Current will flow through the load now when the BASIC Stamp output pin is set Low (ground). Current will flow out of the output pin and up through the load resistor to Vdd. This is the current sinking mode, and the BASIC Stamp can deliver a maximum of 25 mA to the load when configured in this manner.

Output Capability of Digital Circuits

The output capability of digital circuits is listed in the manufacturer's datasheet. Devices usually can "sink" more current than they can "source." Some devices do not have the capability to source current because the internal path from their output to +V is not present. You may see this output design referred to as a device with an "open collector" output.

Outputs have been used to drive LEDs in previous exercises as pictured in Figure 3.2a. When the BASIC Stamp output is low, the diode is forward-biased at approximately one volt, and the remaining four volts, dropped across the 220-ohm resistor, limit current flow to approximately 22 mA. The light emitted by the diode gives visual indication of the output action. In previous programming challenges, you have assumed that the on-off status of an LED represents process action taking place. This is a valid assumption when you consider the operation of a solid-state relay (SSR). Figure 3.2b is a schematic representation of the solid-state relay. The input circuit (terminals 1-2) is equivalent to Figure 3.2a. The +3 to +24 V DC input identifies a range of control voltages. Control voltages must be above the minimum voltage to produce enough LED current for turn-on. Exceeding the maximum control voltage may cause damaging amounts of current to flow in the input LED. The light generated in the SSR strikes an optically controlled output circuit. The detail of this circuit is not shown, but is represented by the normally-open contact symbol. The current and voltage limitations of the output are listed in the device's datasheet and are usually printed on the device itself.

Figure 3.2: BASIC Stamp to LED and Solid-State Relay Schematics

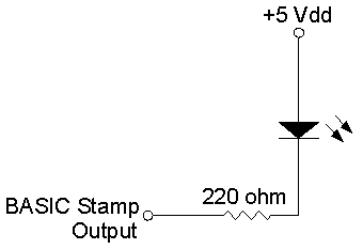


Figure 3.2a: BASIC Stamp to LED

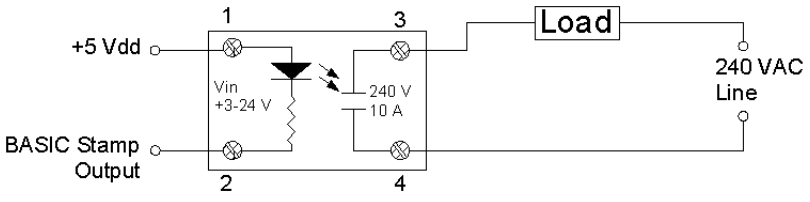


Figure 3.2b: BASIC Stamp to Solid-State Relay

Solid-state relays are available in a wide variety of output ranges. They may be designed to drive either AC loads or DC loads. The load you are driving defines the minimum specification of the SSR required.

An added benefit of the solid-state relay is electrical isolation. The BASIC Stamp is controlling the load by an optically-coupled signal. There is no electrical connection between the microcontroller and the high-power load device. Electrical failures of the load, or power line problems such as spikes, are not fed back to the BASIC Stamp. The datasheet for the Potter-Brumfield SSR may be found in Appendix C. Refer to this datasheet and find the following information.

Input voltage range: _____
Input current requirement @ five volts: _____
Maximum output load current: _____
Maximum output load voltage: _____
Electrical isolation: _____

A word of caution when selecting and implementing solid-state relays:

1. Do not push the specifications to their limits. Oversize the output capability of your selection by at least 20%.
2. Pay close attention to any heatsink requirements. Maximum load current capability is usually dependent upon incorporating a proper heatsink.
3. The load's supply source and all wiring and connections must be able to conduct the load's current. If relays are placed on the breadboard for prototyping, be aware that the breadboard traces are rated at only 1 amp.
4. Respect the output circuit voltage. Be sure all connections are solidly secured and correct before applying line voltage. There is the risk of electrical shock. Take measures to prevent contact with high-voltage potentials. Shield or encase these contacts. Clearly identify high-voltage potentials with appropriate labeling.
5. Some electronic relays will not contain an internal current-limiting resistor on the input. In these cases, an external current-limiting resistor must be added in series with the internal LED. The value of the resistor is based on your control voltage and the manufacturer's recommended input current specification.

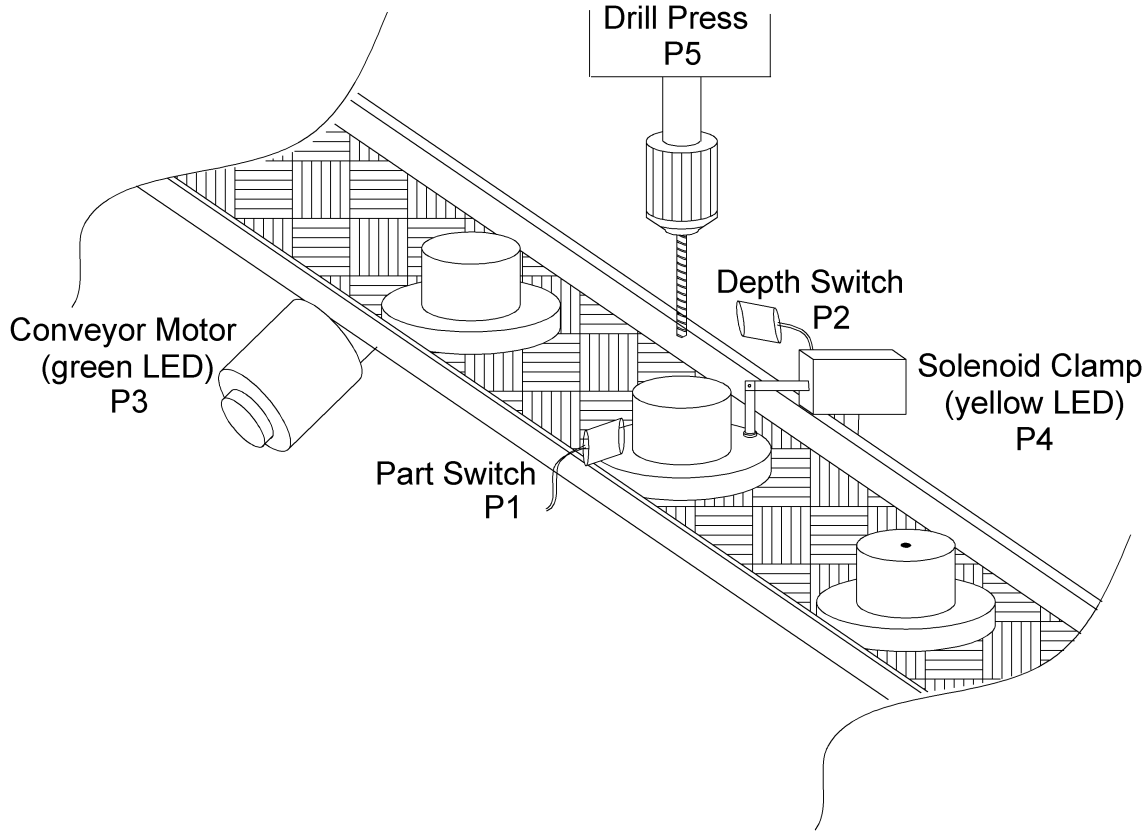
Solid-state relays provide an easy interface for controlling loads in an industrial application. Become familiar with the SSR datasheet specifications in order to make the right selection for your application.

Exercises

Exercise #1: Sequential Control

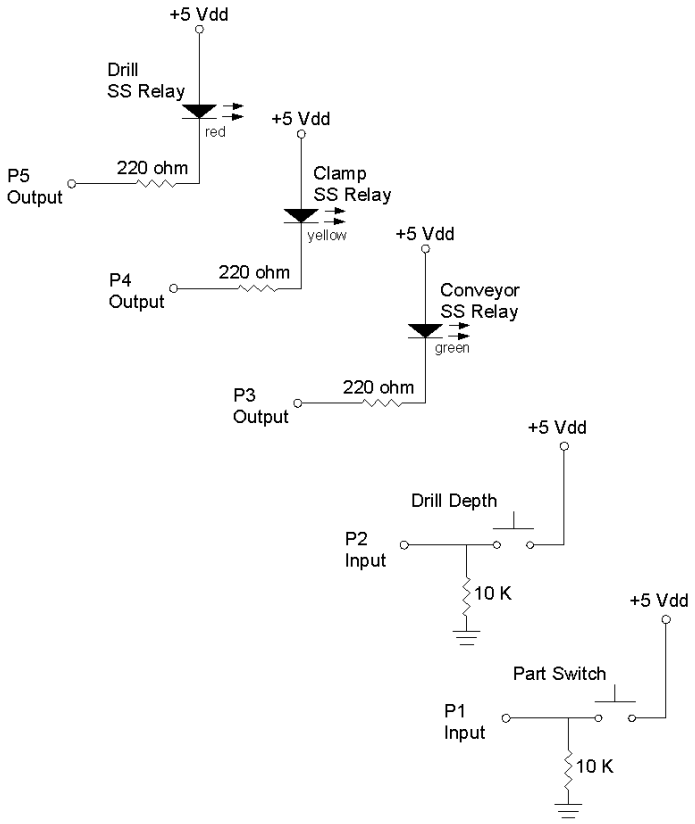
The BASIC Stamp is well suited to perform sequential control operations. Many processes depend on the orderly performance of operations. Consider the machining operation pictured in Figure 3.3a.

Figure 3.3a: AutoDrill Sequence Operation



3

Figure 3.3b: Sequential Control

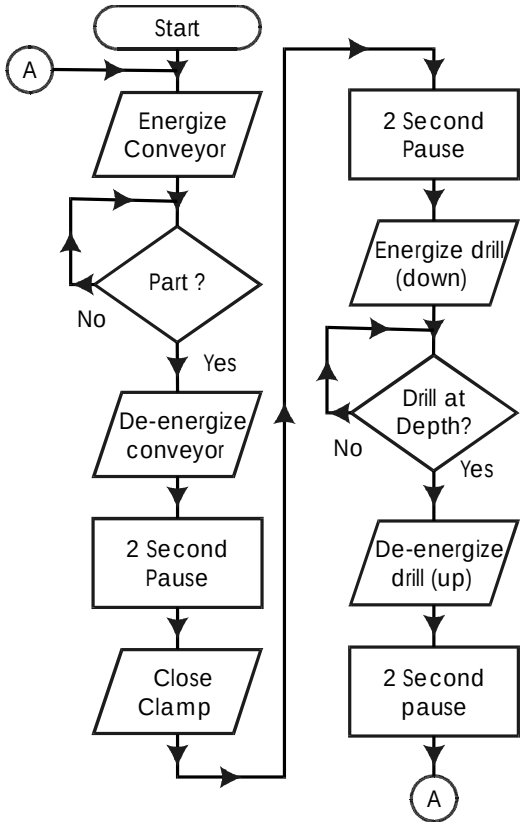


A conveyor is moving parts through a machining station. When a part is detected in the staging area, the conveyor is turned off. After a short pause, the solenoid clamp is activated to hold the part; another short pause, and then the drill is brought down to the part. A proximity switch detects proper depth of the hole. When the depth switch closes, the "drill down" command is stopped and the drill is retracted. After allowing a short time for the drill to retract, the clamp is released and the conveyor is started. This moves the processed part out of the staging area, and the conveyor continues until a new part is detected. Upon detecting another part, the sequential process continues.

With proper signal conditioning, the BASIC Stamp can easily control this sequence. For our exercise purposes, you are asked to use your imagination to allow LEDs to simulate the SSRs that could control the conveyor, clamp, and drill. Two pushbuttons and your two fingers must simulate the part coming into position and the drill coming down to proper depth. Construct Figure 3.3b on your Board of Education. For easier identification, use your green LED for the conveyor, your yellow LED for the clamp, and the red LED for the drill.

Sequential control lends itself well to flowcharting. The time required to develop your flowchart will be quickly saved as you write your program. Compare the flowchart in Figure 3.4 to the description of the machining process.

Figure 3.4: Sequential Control Flowchart



Program 3.1 follows the flowchart very closely. Study the program; compare its structure to the flowchart. Enter the program and try it. Again, you will have to use your imagination to simulate our process.

When the program starts, the green LED will be on. This represents the conveyor starting. To simulate a part coming into the staging area, you must press and hold Pushbutton P1. The conveyor LED will instantly turn OFF and the yellow LED indicating the Clamp relay will turn on after a one-second delay. After the Clamp has had two seconds to secure the part, the drill will come down toward the part as indicated by the red LED. At this time bring another finger down to simulate the drill. Pushbutton P2 represents a proximity switch, which will indicate when proper drill depth has been reached. Your "drill" finger pressing P2 will be turned OFF; the red LED indicating the drill is retracting. Your finger now coming off of the P2 pushbutton indicates the bit has started retracting and two seconds will be allowed for the drill to clear the part. After this delay, the clamp will be opened (yellow light OFF) and the conveyor will start again. The part is completed and leaves the staging area. From this point the sequence starts again.

Run the program a few times. Other than the DEBUG report that a part has been completed, there is no need for your computer. Unplug the serial cable from the Board of Education and continue to simulate the sequential process. The BASIC Stamp could function as the "embedded controller" in this application. Wiring the actual field devices to the BASIC Stamp would allow it to continuously repeat the process.

After understanding this sequential process, we will redefine your two inputs and three outputs to simulate another operation. You will be challenged to develop the program necessary for this embedded control application.

```
'Program 3.1: Sequential Process Control Machining Operation - Embedded

INPUT 1                          ' Part Detection Switch
INPUT 2                          ' Drill Depth Switch
OUTPUT 3                         ' Conveyor motor relay (green)
OUTPUT 4                         ' Clamp solenoid relay  (yellow)
OUTPUT 5                         ' Drill press relay (red)

OFF con 1                        ' Current sink loads
ON  con 0                        ' Negative logic

OUT3 = OFF                       ' Initialize outputs off
OUT4 = OFF
OUT5 = OFF
Start:
  OUT3 = ON                      ' Conveyor on
  IF IN1 = 1 THEN Process        ' If pressed, start "Process"
```

```
GOTO START

Process:                          ' The process begins
  OUT3 = OFF                      ' Stop conveyor
  PAUSE 1000
  OUT4 = ON                       ' Begin clamping part in place
  PAUSE 2000                      ' Wait 2 seconds to turn drill on

Drill_down:
  OUT5 = ON                       ' Turns on drill and drill begins dropping
  IF IN2 = 1 Then Pull_drill      ' If drill is deep enough, pull drill
GOTO Drill_down

Pull_drill:
  OUT5 = OFF                      ' Turns off drill and drill retracts
  IF IN2 = 0 Then Drill_up        ' Indicates drill is moving up
GOTO Pull_drill

Drill_up:
  PAUSE 2000                      ' Continue pulling drill up for 2 seconds

Release:
  OUT4 = OFF                      ' Open clamp to release part
  PAUSE 1000                      ' Wait 1 second
  OUT3 = ON                       ' Conveyor on
  IF IN1 = 0 Then Nextpart        ' Finished part leaves process area
GOTO Release

Nextpart:
  PAUSE 1000                      ' Wait 1 second
  DEBUG "Part leaving clamp. Starting next cycle", CR
GOTO Start
```

The real beauty of microcontrollers is to have the capability of embedding all of the intelligence necessary to perform sophisticated control within the equipment.

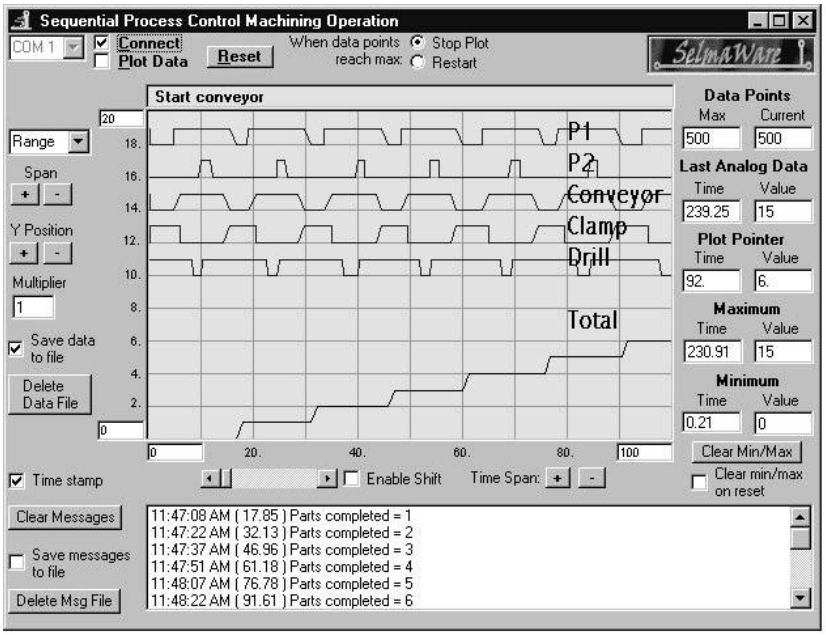
There are times, however, that being able to retrieve information from the microcontroller adds to its capabilities. The StampPlot Lite interface can be effectively used to monitor the sequential machining process. Program 3.2 uses this interface. The machine functions in Program 3.2 are the same as those in the previous program. **Debug** commands have been embedded to send data to the StampPlot Lite interface. The program plots the status of the digital I/O, reports process steps in the User status bar, and keeps a time-stamped list of the total parts produced. Figure 3.5 is a representative screen shot of the sequential process being monitored by StampPlot Lite. Load Program 3.2 and run it. Study the StampPlot Lite **debug** commands that have been added to the original program. Become familiar with their use. Graphical user interfaces such as this are very useful in the maintenance and data acquisition of embedded control systems. Use StampPlot to monitor the Sequential Control Mixing Challenge at the end of this section.

```
Program 3.2: Sequential Process Control Machining Operation with StampPlot
Interface

Pause 500
DEBUG "!TITL Sequential Process Control Machining Operation", 13
' StampPlot title
DEBUG "!TMAX 100", 13                  ' Set sweep plot time (seconds)
DEBUG "!PNTS 500", 13                  ' Sets the number of data points
DEBUG "!AMAX 20",  13                  ' Sets vertical axis (counts)
DEBUG "!CLRM", 13                      ' Clear List Box
DEBUG "!CLMM", 13                      ' Clear Min/Max
DEBUG "!RSET", 13                      ' Reset all plots
DEBUG "!DELD", 13                      ' Delete old data file
DEBUG "!PLOT ON", 13                   ' Turn Plot on
DEBUG "!TSMP ON", 13                   ' Time-stamp part completion
DEBUG "!SAVD ON", 13                   ' Save data to file

INPUT  1                               'Part Detection Switch
INPUT  2                               'Drill Depth Switch
OUTPUT 3                               'Conveyor motor relay (green)
OUTPUT 4                               'Clamp solenoid relay (yellow)
OUTPUT 5                               'Drill press relay    (red)

OFF con 1                              'Current sink mode
ON  con 0                              'Negative logic

OUT3 = OFF                             ' Initialize outputs off
OUT4 = OFF
OUT5 = OFF

Parts var byte
Parts = 0
```

```
Start:
  GOSUB Plot_data                        ' Plot the status
  OUT3 = ON                              ' Conveyor on
  DEBUG "!USRS Start conveyor",13        ' User status prompt
  IF IN1 = 1 THEN Process                ' If pressed, start "Process"
  PAUSE 100
GOTO START

Process:                                 ' The process begins
  GOSUB Plot_data                        ' Plot the status
  OUT3 = OFF                             ' Stop conveyor
  DEBUG "!USRS Detected part. Stop conveyor",13      ' User status prompt
  PAUSE 1000

  GOSUB Plot_data                        ' Plot the status
  OUT4 = ON                              ' Begin clamping part in place
  DEBUG "!USRS Clamp part.",13           ' User status prompt
  GOSUB Plot_data                        ' Plot the status
  PAUSE 2000                             ' Wait 2 seconds to turn drill on

Drill_down:
  GOSUB Plot_data                        ' Plot the status
  OUT5 = ON                              ' Turns on drill and drill drops
  DEBUG "!USRS Drill coming down!",13 ' User status prompt
  IF IN2 = 1 Then Pull_drill             ' If drill is deep enough, pull drill
  PAUSE 100
GOTO Drill_down

Pull_drill:
  GOSUB Plot_data                        ' Plot the status
  OUT5 = OFF                             ' Turns off drill and drill retracts
  DEBUG "!USRS Stop Drill and Retract",13
                                         ' User status prompt
  IF IN2 = 0 Then Drill_up               ' Indicates drill is moving up
  PAUSE 100
GOTO Pull_drill

Drill_up:
  GOSUB Plot_data                        ' Plot the status
  DEBUG "!USRS Drill coming up!!",13     ' User status prompt
  PAUSE 2000                             ' Pull drill for 2 seconds

Release:
  GOSUB Plot_data                        ' Plot the status
  OUT4 = OFF                             ' Open clamp to release part
  DEBUG "!USRS Clamp released. Conveyor moving.",13
'User status prompt
```

```
  PAUSE 1000                    ' Wait 1 seconds
  OUT3 = ON                     ' Conveyor on
  IF IN1 = 0 Then Nextpart
  GOTO Release

Nextpart:
  GOSUB Plot_data               ' Plot the status
  DEBUG "!USRS Part Complete. Start next cycle",13
' User status prompt
  Parts = Parts + 1             ' Parts counter
  PAUSE 1000                    ' Wait 1 seconds
  DEBUG "Parts completed = ", DEC Parts,13
                                ' Post parts count in the List Box
GOTO Start

Plot_data:
  DEBUG IBIN IN1,BIN IN2,BIN OUT3,BIN OUT4, BIN OUT5,13
'Plot the digital status.
  DEBUG DEC Parts,13            'Plot analog count
RETURN
```

Figure 3.5: Screen Shot of the Sequential Machining Process using StampPlot Lite



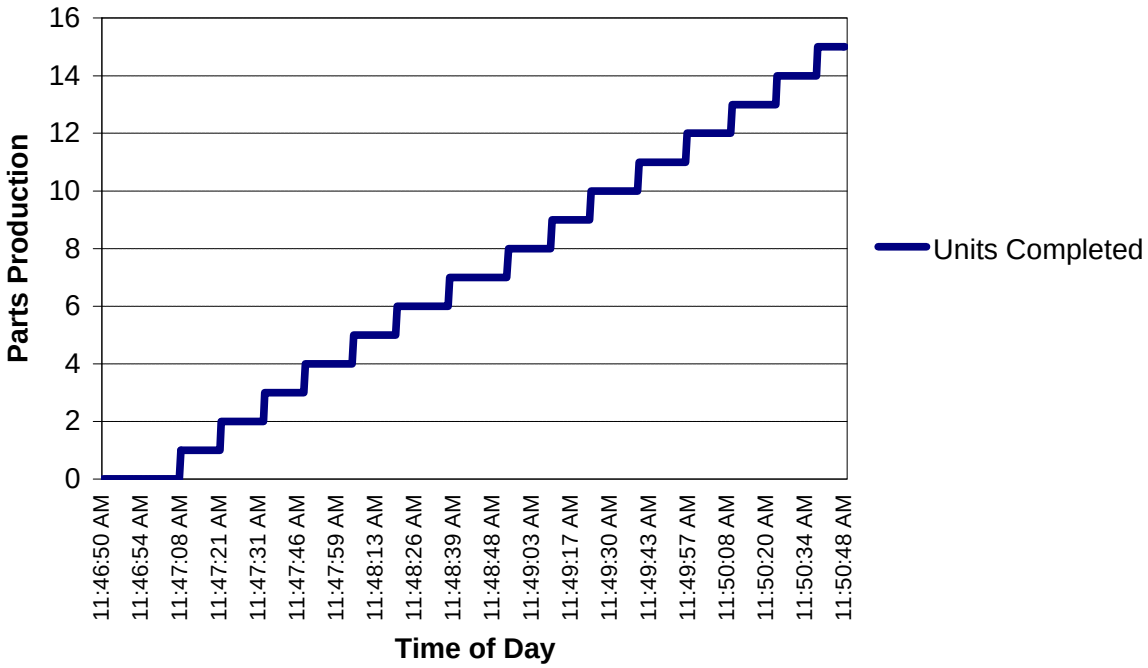
Note that the traces appear from top to bottom in the order which they were listed in the Debug digital plot command. Therefore, the top two traces are of the active high pushbuttons **IN1** (product in position) and **IN2** (depth switch). The next three traces are outputs **OUT3** (conveyor), **OUT4** (clamp), and **OUT5** (drill). Remember that the outputs are wired in the current sink mode. A High is OFF and a Low is ON.

Notice that in the initial setting for the StampPlot Lite interface, "Save data to file" (!SAVD) is ON. During the production run, the data at each sample point is saved into a text file, stampdat.txt. The data includes the time of day and program time that the sample was taken, the sample number, and the analog and digital values at the time of each sample. The data are comma delimited (separated by commas), and therefore, ready to be brought into a variety of spreadsheet or database software packages. Once the data is in the package, it is available for analysis and manipulation. Figure 3.6 represents a portion of the production run data, as it would appear in a Microsoft Excel spreadsheet. The complete file contains 500 samples (rows of data). Figure 3.7 is an Excel graph constructed from the data file.

Figure 3.6: Sequential Control Production Run (samples only)

Time of Day	Run Time	Sample number	Units Completed	Sample number	Digital Status
11:46:50 AM	0.21	1	1	1	111
11:46:50 AM	0.21	2	2	2	11
11:46:50 AM	0.21	3	3	3	11
11:46:50 AM	0.21	4	4	4	11
11:46:50 AM	0.27	5	5	5	11
11:46:50 AM	0.27	6	6	6	11
11:46:50 AM	0.27	7	7	7	11
11:46:50 AM	0.27	8	8	8	11
11:46:50 AM	0.27	9	9	9	11
11:46:50 AM	0.27	10	10	10	11
11:46:50 AM	0.32	11	11	11	11
11:46:50 AM	0.32	12	12	12	11
11:46:51 AM	0.43	13	13	13	11
11:46:51 AM	0.50	14	14	14	11
11:46:51 AM	0.71	15	15	15	11
11:46:51 AM	0.98	16	16	16	11
11:46:51 AM	1.26	17	17	17	11

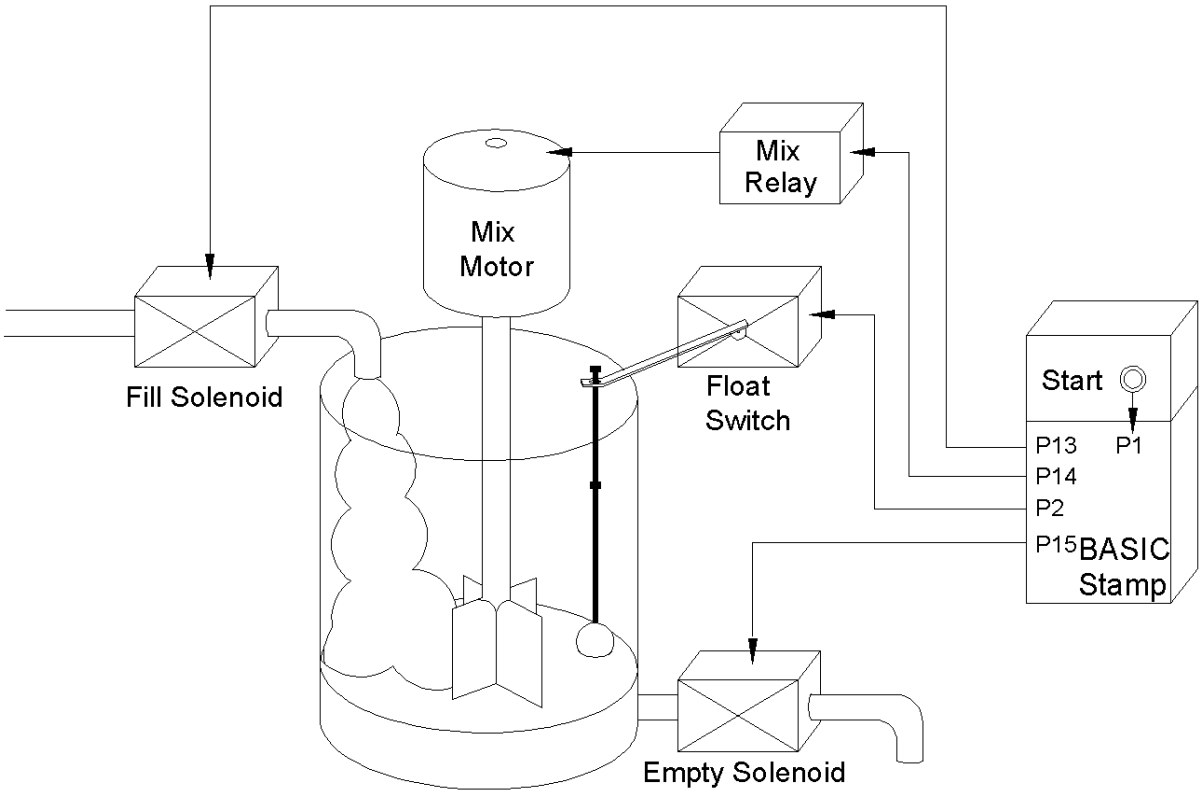
Figure 3.7: Graph of Sequential Control Production Run



Programming Challenge: Sequential Mixing Operation

A mixing sequence is pictured in Figure 3.8. In this process, an operator momentarily presses a switch to open a valve and begin filling a vat. A mechanical float rises with the liquid level and closes a switch when the vat is full. At this time, the "fill" solenoid is turned off, and a mixer blends the vat contents for 15 seconds. After the mixing period, a solenoid at the bottom of the vat is opened to empty the tank. The mechanical float lowers, opening its switch when the vat is empty. At this point, the "empty" solenoid is turned off and the valve closes. The process is ready for the operator to start another batch.

Figure 3.8: Mixing Sequential Control Process



Assign the following to the BASIC Stamp inputs and outputs to simulate the operation.

Operator pushbutton	Input P1 (N.O. active high)
Float switch	Input P2 (N.O. active high)
Fill Solenoid	Output P13 (red LED)
Mix Solenoid	Output P14 (yellow LED)
Empty Solenoid	Output P15 (green LED)

Construct a flowchart and program the operation.

Exercise #2 Current Boosting the BASIC Stamp

The BASIC Stamp's output current and/or voltage capability can be increased with the addition of an output transistor. Consider the circuit in Figure 3.9a. The circuit values should be designed such that a high (+5V) output of the BASIC Stamp drives Q1 into saturation without drawing more current than the BASIC Stamp can source.

Figure 3.9: Sequential Machining Process Schematic

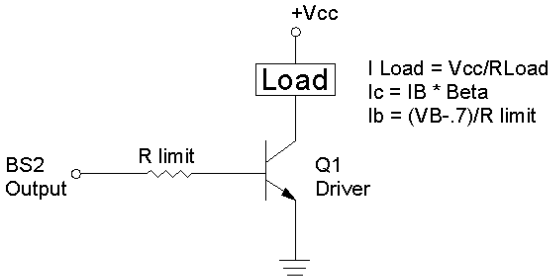


Figure 3.9a: BASIC Stamp to Transistor Driver

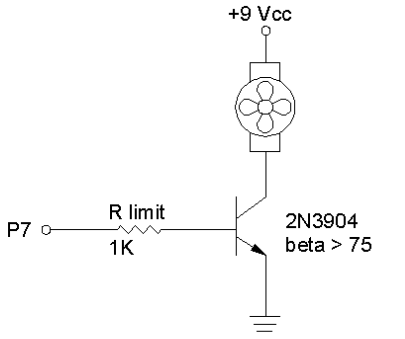


Figure 3.9b: BASIC Stamp Driving the Fan

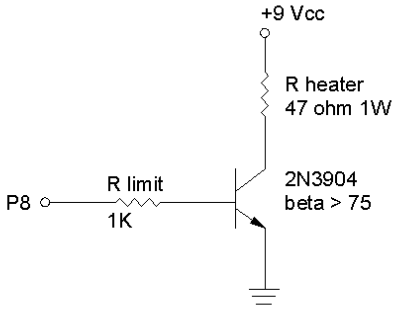


Figure 3.9c: BASIC Stamp Driving the Heater

Circuit component values stem from the load current and voltage requirements. The process of determining minimum component values is as follows: Since Q1 acts as an open collector current sink to the load, the load's supply voltage is not limited to the BASIC Stamp's +5-volt supply. If separate supplies are used, however, their common ground lines must be connected. When Q1 is driven into saturation, virtually all of the supply voltage will be dropped across the load and the load current will be equal to Vss/R_{load}. Q1's maximum collector current capability must be higher than this load current. The Q1 base current required to yield the collector current may be calculated by dividing the load current by the "beta" of Q1. $Ib = Ic/\beta_{Q1}$. Given a 20 mA maximum BASIC Stamp output current, a minimum transistor beta may be calculated by rearranging this formula. $\beta_{Q1(min)} = Ic/Ib$. Where Ib is the 20 mA maximum BASIC Stamp drive current.

A transistor must be chosen that meets, and preferably exceeds, these minimum requirements. Exceeding the minimum values by 50 to 100% or more would be best. Once the transistor is chosen, an appropriate base-limiting resistor value can be determined. This value must allow more base current than that defined by $Ic/\beta_{Q1,}$ yet less than the 20 mA BASIC Stamp limit. The voltage drop across R_{limit} is equal to the +5 V BASIC Stamp output minus the PN junction drop of Q1 (approximately +5V-.7V, or 4.3V).

For this exercise, and upcoming experiments, we have two loads that we wish to drive in this manner. They are a brushless DC fan and a 47-ohm, half-watt resistor. The brushless fan specifications include a full line voltage of +12 V and line current of 100 mA. The resistor will draw approximately 190 mA when powered by the +9 V Vin power supply.

Following the procedure outlined above, the transistors must handle collector currents of at least 190 mA and have a beta specification greater than 10. Figures 3.9b and 3.9c represent a 2N3904 as Q1. This transistor has collector current capability of 300 mA and a minimum Beta of 75. Added features to the selection of this transistor include that it is very common, it is inexpensive, and it can deliver the load current without need of a heat sink. R_{limit} values were selected based on minimum beta specifications and a desire to keep the BASIC Stamp output current demand well below 20 mA. These 1K-ohm resistors allow approximately 5 mA of base current, which ensures saturation.

Construct the two transistor-driver circuits in Figure 3.9b and 3.9c on your Board of Education. To test the devices, enter the short Program 3.3 to configure Pin 7 and Pin 8 as outputs, and turn them on.

```
Program 3.3: Current-boost for the fan and heater.

Output 7            ' Output for Fan drive
Output 8            ' Output for Heater drive
Out7 = 1            ' Fan ON (active high)
Out8 = 1            ' Heater ON (active high)
```

The fan will be running at full speed, and the resistor will be warming up due to the current flowing through it. Use your multi-meter to measure the collector emitter voltage of each transistor. If they are fully saturated, this voltage should measure less that 300 mV. If the transistors are not fully saturated, the R_{limit} value can be reduced.

In the next experiment, we will use the resistor to simulate a heating element. The fan will simulate a process disturbance that cools the heater. Our objective will be to investigate various types of control to maintain a constant temperature. Leave these circuits constructed on your Board of Education.

Before we leave this exercise, it is worth mentioning some other interfacing challenges that you may be confronted with as a designer. Consider the circuits in Figure 3.10.

Figure 3.10: BASIC Stamp Output Interfacing

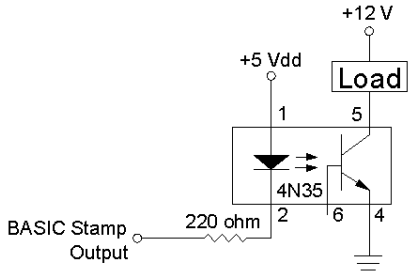


Figure 3.10a: BASIC Stamp to Optocoupler

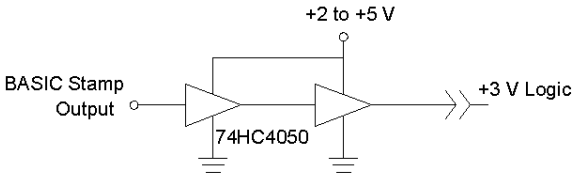


Figure 3.10b: BASIC Stamp to HCMOS

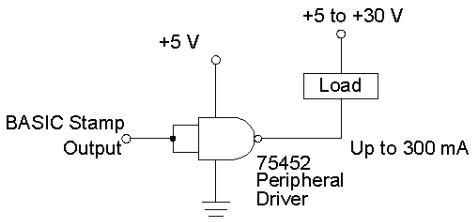


Figure 3.10c: BASIC Stamp to High Current Driver

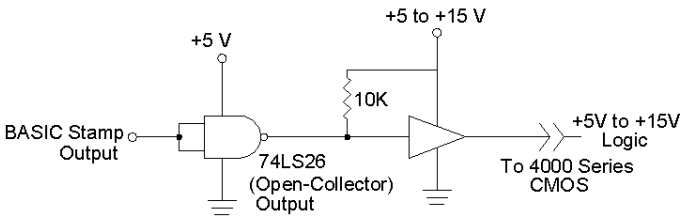


Figure 3.10d: BASIC Stamp to +15 V Logic

(a) The opto-coupler can be used to interface different voltages and to electrically isolate an output from the microcontroller circuit in Figure 3.10a.

(b) Figure 3.10b can be used to interface to HCMOS or 4000-series CMOS devices. The 74HC4050 can be operated on low voltages, allowing interfacing to +3-volt logic.

(c) There is a large variety of peripheral driver chips available. The 75452 driver depicted in Figure 3.10c can sink up to 300 mA of load current. Its open-collector output allows for loads up to 30 volts.

(d) Figure 3.10d includes the 74LS26 NAND gate. This is one of a family of open-collector gates. With the 10K-ohm pull-up resistor referenced to the next circuit stage, the BASIC Stamp can be interfaced to higher-voltage CMOS circuits.

Questions and Challenge

Questions

1. Output field devices are those devices that do the _____ in a process control application.

2. Field devices usually require more power than the BASIC Stamp can deliver. List three power interface devices that can control high-power circuits and be turned on and off by the BASIC Stamp.
   a. _____
   b. _____
   c. _____

3. The BASIC Stamp output is acting as a current sink when the load it is driving is connected between the output pin and _____.

4. The BASIC Stamp can source _____mA per output.

5. Electronic and electromagnetic relays offer a level of protection to the microcontroller because they provide electrical _____ between the BASIC Stamp and the power devices.

6. The input circuit of an SSR is usually an _____ ,which provides light that optically triggers an output device.

7. The current rating of an SSR should be oversized by at least _____ percent of the continuous load current demand.

8. Maximum continuous current ratings of solid-state relays usually involve applying a _____ for proper heat dissipation.

9. _____ control involves the orderly performance of process operations.

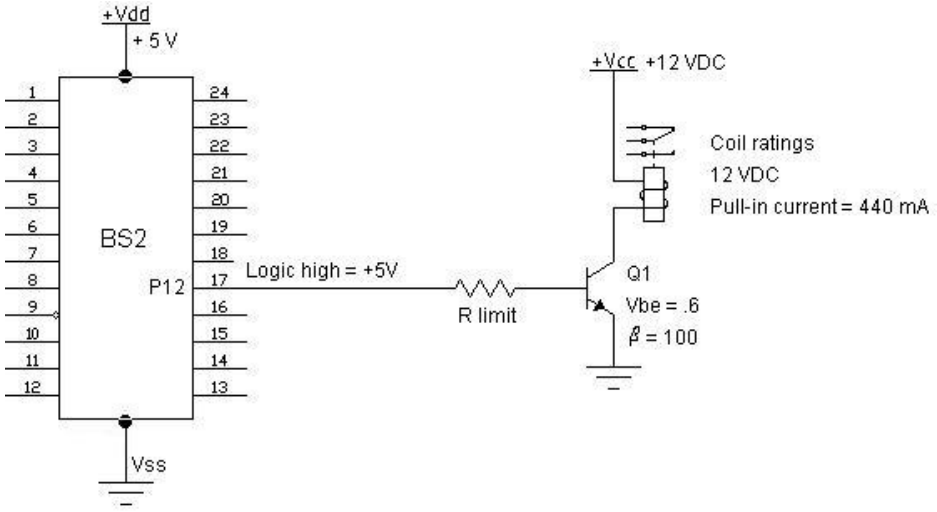
10. When the output current from the BASIC Stamp is not sufficient to turn on the control device, an output _____ may be used for current boosting.

11. The contacts of an electromagnetic relay are shown in schematics in the "normal" position. Normal means the relay's coil is _____ energized.

12. The "contacts" of an AC solid-state relay are actually the main terminals of a TRIAC. These contacts would be depicted in a schematic as being normally _____.

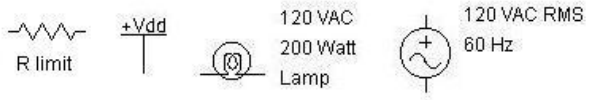
Design It!

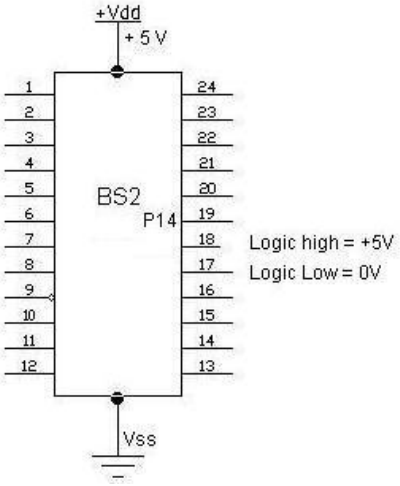
1. Given the figure below, solve for the maximum value of the base limiting resistor (Rlimit) that would allow the 440 mA of coil current to flow when the BASIC Stamp output Pin 12 is high.

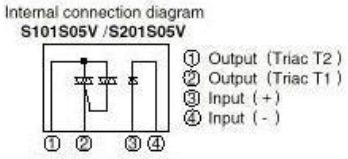


2. To ensure deep saturation of transistor Q1, the value of Rlimit should be _____ than this value.

3. The internal connection diagram of the SHARP S101S05V solid-state relay is given below. Notice that its input circuit is just an LED. The datasheet specifies that the LED has a forward voltage drop of 1.2 volts and that 15 mA through the LED will turn on the relay. Use the following components to complete the diagram for controlling the SSR with Pin 14 of the BASIC Stamp. Configure it as a current sink. Calculate the proper value of Rlimit. Draw the lamp and 120 VAC source as they would be connected to the SSR outputs.



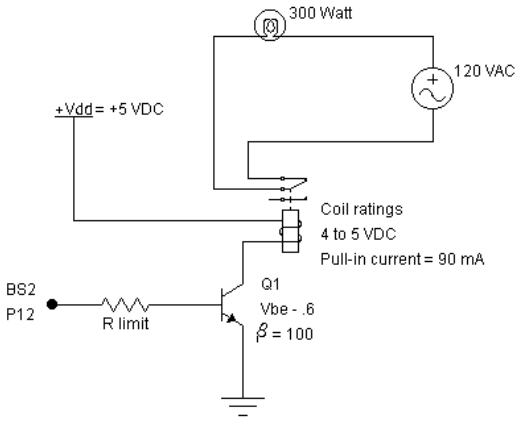




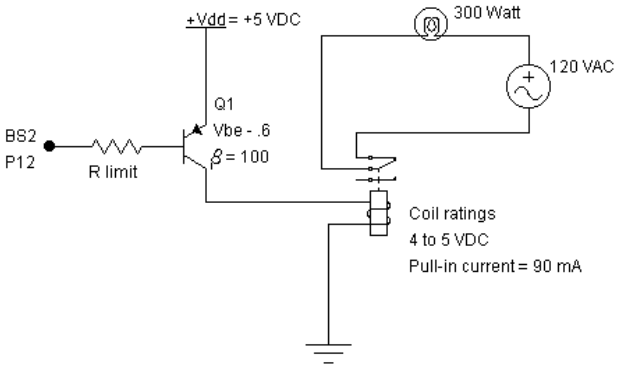
Analyze it!

1. Consider circuits A and B below. Write a line of BASIC Stamp code that will result in turning the lamp ON for each.

   Circuit A _____. Circuit B _____.



Circuit A: Current source drive and normally closed contacts.



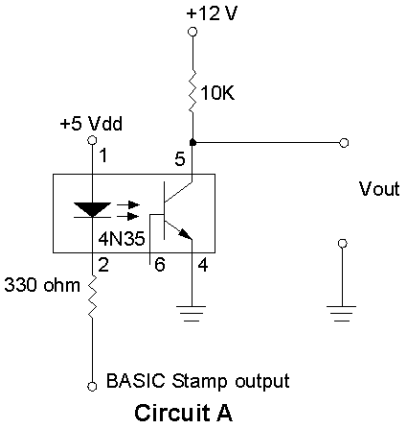
Circuit B: Current sink drive and normally open contacts.

2. Study the three figures shown below. Would you write a logic High or a logic Low to the BASIC Stamp output to yield a 12-volt Vout value?

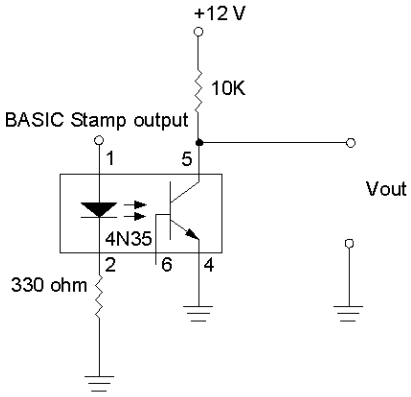
   Circuit A _____

   Circuit B _____

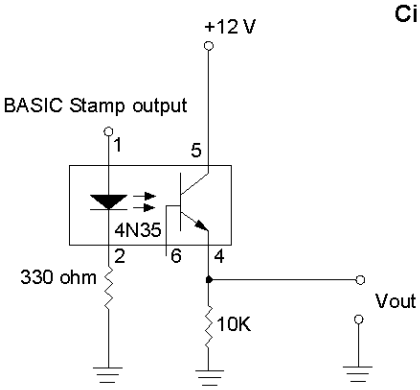
   Circuit C _____



Circuit A



Circuit B



Circuit C

Program it!

1. Given the input and outputs pictured back in Figure 3.3b, write a sequential program that will do the following:

   Momentarily pressing P1 will cause P3 to turn ON for three seconds and then go OFF. Pressing P1 a second time will cause P4 to come ON for three seconds and then go OFF. When P4 goes OFF, P5 will come ON until P2 is pressed.

2. Try this one. Using the same I/O, write a program that will do the following:

   Press and hold P1 and P3 goes ON. Holding P1 and pressing P2 causes P3 to go OFF and P4 to come ON. Releasing P1 while continuing to hold P2 turns OFF P4 and ON P5. And lastly, releasing P2 will turn all three outputs ON for three seconds, then all OFF, and the process is set to repeat.